Integer Solutions to Cutting Stock Problems

L. Fernández, L. A. Fernández, C. Pola

Dpto. Matemáticas, Estadística y Computación, Universidad de Cantabria, 39005 Santander, Spain,
laura.fernandezfern@alumnos.unican.es, lafernandez@unican.es, polac@unican.es

Abstract
We consider two integer linear programming models for the one-dimensional cutting stock problem that include various difficulties appearing in practical real problems. Our primary goals are the minimization of the trim loss or the minimization of the number of master rolls needed to satisfy the orders. In particular, we study an approach based on the classical column-generation procedure by Gilmore and Gomory for solving the linear programming (LP) relaxations and an extra column-generation procedure before solving a final integer problem. For our computational tests we use some data sets from the paper industry and some others generated randomly.
Keywords: One-dimension cutting stock, integer solutions, knapsack problem.

1. Introduction

This work deals with the real-world industrial problem of reel cutting optimization, usually called Cutting Stock Problem (CSP) that can be described as follows: find the most economical way of cutting the master rolls (with large width) into rolls with smaller widths previously specified by customers together with the quantity they need. It is well known that several formulations are possible (see for example [1]). Here, we consider two integer linear programming models for the one-dimensional CSP differing in the objective cost function: in the Model I, our primary goal is the minimization of the trim loss and in the Model II, we consider the minimization of the number of master rolls needed to satisfy the orders. This last model has been used more frequently and has some interesting features like the modified integer round-up property. Theoretically, both models are equivalent when the demands must be satisfied exactly. Nevertheless, in practical real problems the demands usually may be over satisfied, because the customer accepts certain deviations of the quantity ordered. These conditions of sale can be agreed upon the client and the seller in the contract, but there exist some standard tolerances (see the "General Conditions of Sale of Paper and Board Manufacturers in EEC", by the European Confederation of Pulp, Paper and Board industries (CEPAC) a document produced in 1991, but still in use). Other practical difficulties usually appearing are: the limitation on the maximum waste allowed in each cutting pattern (i.e. list of items to be cut from a master roll) and a limited number of slitting knives. In particular, we study an approach based on the classic column-generation procedure by Gilmore and Gomory [6] for solving the LP relaxations (without integer variables) of the integer CSP. The LP optimal solution, which generally is non-integer, has associated a small fraction of all possible patterns. To obtain an integer solution we solve a final integer CSP after using an extra column-generation procedure.
For our computational tests we use some data sets from the paper industry and others randomly generated with the problem generator CUTGEN1, see [4]. In the first case we obtain a very good performance in terms of paper savings when compared with those solutions obtained by the paper mill. The models are compared with respect to solution quality and computing time.
The work presented here corresponds to a first stage of a project with a paper company whose last interest was to produce an interactive system able to solve this problem (with several adjustable parameters) for running on a personal computer. The use of a second machine (rewinder) that allows making further cuts on the rolls initially produced on the first machine (winder) will be considered in another paper.
On the other hand, the main results of this study are included in the undergraduate academic work [3], where more details can be found.
According to the typology of Wäscher et al. [11], the problem under consideration is of type SSSCSP (Single Stock Size Cutting Stock Problem).

2. Problem formulation

Let us present the main characteristics and data of our problem that are relevant to arrive to its mathematical formulation:

1. All the master rolls have the same length of paper.
2. The cuts are produced on the winder with a limited number N of knives.
3. Minimum (W_{min}) and maximum (W_{max}) widths are imposed to cutting patterns.
4. The total number n of finished roll types required and their widths $w_i,\ i = 1, \ldots, n$ are known.
5. The over-production for each item i is limited with a tolerance. The lower and upper bounds on the number of finished rolls d_i and D_i are also data, for each i. It can happen that $d_i = D_i$.

If we denote by a_{ij} the number of final rolls of width w_i to be slit from each master roll that is processed by using pattern j, any cutting pattern may be represented by a column vector, $(a_{1j}, a_{2j}, \ldots, a_{nj})^T$, satisfying the following conditions:

$$\begin{cases} W_{min} \le \sum_{i=1}^{n} w_i a_{ij} \le W_{max}, \\ \sum_{i=1}^{n} a_{ij} \le N - 1, \\ a_{ij} \ge 0 \ \text{ and integer}, \quad i = 1, ..., m, \end{cases} \tag{1}$$

Finally, denoting the total number of patterns for a given problem by m, the number of master rolls to be slit with pattern j by x_j and the trim loss incurred by pattern j by c_j, the problem for Model I can be formulated as follows:

$$\begin{cases} \text{Minimize} & f(x) = c^T\ x \\ \text{subject to} & d_i \le \sum_{j=1}^{m} a_{ij} x_j \le D_i, \quad i = 1, ..., n, \\ & bl \le \sum_{j=1}^{m} x_j \le bu, \\ & x_j \ge 0 \ \text{ and integer}, \quad j = 1, ..., m, \end{cases} \tag{2}$$

where $c = (c_1, \ldots, c_m)^T$ and bl, bu are the lower and upper bounds (resp.) on the number of processed master rolls. We can rewrite (2) as follows:

$$\begin{cases} \text{Minimize} & f(x) = c^T\ x \\ \text{subject to} & B\ x \ge h, \\ & x \ge 0 \ \text{ and integer}, \end{cases} \tag{3}$$

where the notation $x \ge 0$ means $x_i \ge 0$ for $i = 1, \ldots, m$,

$$B = (b_{ij}) = \begin{pmatrix} A \\ -A \\ e \\ -e \end{pmatrix}, \quad h = \begin{pmatrix} d \\ -D \\ bl \\ -bu \end{pmatrix} \tag{4}$$

with $A = (a_{ij})$, $d = (d_1, \ldots, d_n)^T$, $D = (D_1, \ldots, D_n)^T$ and $e = (1, \ldots, 1) \in \mathbb{R}^m$.
Let us remark that the costs associated with pattern changes are not taken into account: this is not relevant from the company's viewpoint because their machines made the changes automatically and quite fast.
For Model II (minimizing the number of used master rolls) the unique modification is that $c_j = 1$ for all $j = 1, \ldots, m$.

3. Overview of the algorithm

The problems occurring in the paper industry may involve millions of potential cutting patterns (see for instance [9]). So it is not practical to generate the whole pattern matrix A to solve the problem. Instead of that we will consider three column generation phases. The two first phases are related with the continuous relaxation problem and the last one concerns to the gap between the continuous and the integer problem.

3.1. Column generation. Phases I and II

Here we consider the continuous relaxation of problem (2) by discarding the integrality constraints on the variables:

$$\begin{cases} \text{Minimize} & f(x) = c^T\ x \\ \text{subject to} & B\ x \geq h, \\ & x \geq 0. \end{cases} \tag{5}$$

An optimal solution of this LP problem, x^{LP}, can be obtained by using a column-generation procedure developed by Gilmore and Gomory [6]. Note that $f(x^{LP})$ is a lower bound on $f(x^{IP})$ for all x^{IP} optimal solution of (2). The bound $f(x^{LP})$ is very tight for Model II: it has been conjectured (see [10]) that the gap between $f(x^{IP})$ and $f(x^{LP})$ is always smaller than 2.
To outline the column generation approach we consider the following theorem about the optimality conditions of the LP problem (see [2] for instance).

Theorem 1 (Kuhn-Tucker conditions) *A feasible solution $x^{LP} \in \mathbb{R}^m$ of (5) is optimal if and only if there are (Lagrange multiplier) vectors $\lambda \in \mathbb{R}^{2n+2}$ and $\mu \in \mathbb{R}^m$ such that*

$$\begin{array}{ll} c - B^T\ \lambda - \mu = 0, & \\ \lambda_i \geq 0 \quad \textit{and} \quad \lambda_i \left(\displaystyle\sum_{j=1}^{m} b_{ij}\ x_j^{LP} - h_i \right) = 0, & i = 1, \ldots, 2n+2, \\ \mu_j \geq 0 \quad \textit{and} \qquad\qquad \mu_j x_j^{LP} = 0, & j = 1, \ldots, m. \end{array} \tag{6}$$

Column generation starts with a set of a few patterns (enough to satisfy the demands). Associated to each subset of patterns there is a submatrix $\bar{B}$ of B, and the corresponding restricted problem (5). Denoting by $\bar{x}^R$ a solution of one of those restricted problems and extending to a feasible point of the complete problem by adding zeros, we use the above theorem for determining if an optimal LP solution of problem (5) has been obtained. The conditions associated with the restricted problem are verified for some $\bar{\lambda} \in \mathbb{R}^{2n+2}$ and $\bar{\mu} \in \mathbb{R}^{m_R}$ where m_R is the number of patterns in the restricted problem. The other optimality conditions are associated with the patterns that are not included in the restricted problem. The question is if there exists at least one of those feasible patterns $(a_{1s}, \ldots, a_{ns})^T$ violating the corresponding optimality condition, that is to say:

$$\mu_{m_R+1} = \bar{c}_{m_R+1} - (a_{1s}, \ldots, a_{ns}, -a_{1s}, \ldots, -a_{ns}, 1, -1)\ \bar{\lambda} < 0, \tag{7}$$

where $\bar{c}_{m_R+1}$ denotes the cost of the pattern (its trim loss for Model I and 1 for Model II).
If such a pattern does not exist, we can conclude that $\bar{x}^R$ yields (after completing with zeros) to an optimal solution of the complete LP problem. Otherwise we seek a feasible pattern that maximizes the violations of the associated optimality condition solving the following knapsack problem

$$\begin{cases} \text{Maximize} & \displaystyle\sum_{i=1}^{n} \pi_i\ a_{is} \\ \text{subject to} & W_{min} \leq \displaystyle\sum_{i=1}^{n} w_i\ a_{is} \leq W_{max}, \\ & \displaystyle\sum_{i=1}^{n} a_{is} \leq N - 1, \\ & 0 \leq a_{is} \leq u_i \quad \text{and integer}, \quad i = 1, \ldots, n, \end{cases} \tag{8}$$

where

$$\pi_i = \begin{cases} (\bar{\lambda}_i - \bar{\lambda}_{n+i}) & \text{for Model I}, \\ (w_i + \bar{\lambda}_i - \bar{\lambda}_{n+i}) & \text{for Model II}, \end{cases}$$

and $u_i = min\left\{ D_i, N-1, \left\lfloor \dfrac{W_{max}}{w_i} \right\rfloor \right\}$.

3.2. Solving Knapsack problems

As we have just seen, each pattern generated in Phase II of Column Generation is obtained by solving a multidimensional Knapsack problem with a cardinality constraint of type

$$\left\{\begin{array}{ll}\text{Maximize} & g(y) = \pi^T\ y \\ \text{subject to} & W_{min} \leq w^T y \leq W_{max}, \\ & \displaystyle\sum_{i=1}^{n} y_i \leq N-1, \\ & 0 \leq y_i \leq u_i \quad \text{and integer}, \quad i = 1, \dots, n,\end{array}\right. \tag{9}$$

where the coefficients of the objective function depend on the model of cutting stock problem. Without loss of generality, here we assume that $\pi_i > 0$ and variables are sorted by decreasing efficiencies such that

$$\frac{\pi_1}{w_1} \geq \frac{\pi_2}{w_2} \geq \dots \geq \frac{\pi_n}{w_n}. \tag{10}$$

Let us denote by $\mathcal{A}$ the set of feasible points for (9), that we are assuming non-empty, and $\bar{y}$ an optimal solution. To solve the integer Knapsack subproblems we have implemented our version of the Gilmore and Gomory [6] branch and bound algorithm, as described by Chvátal [2], by using our own upper bounds that improve the ones by Martello and Toth [8].

Branch and bound mechanism is based on an intelligent enumeration of the feasible points. The idea is to select only a small subset of the feasible points that contains at least one solution of the problem. The following results help us to save computations. In the first result, it is stated that by increasing any component of an optimal solution yields to an infeasible point.

Proposition 1 *Every optimal solution of (9) satisfies at least one of the following conditions:*

$$\sum_{i=1}^{n} \bar{y}_i + 1 > N - 1,$$

$$\sum_{i=1}^{n} w_i\ \bar{y}_i + w_k > W_{max}, \quad k = 1, \dots, n,$$

$$\bar{y}_k + 1 > u_k, \quad k = 1, \dots, n.$$

Feasible points verifying one of the above conditions will be referred to as critical points. Essentially our method for solving the knapsack problems relies on generating critical points. We reduce the number of points to be explored by using the following results (see [2]):

Proposition 2 *Let y denote a feasible point for (9) and $t \in \{1, \dots, n-1\}$ be one of its coordinates. Then, it holds*

$$g(y) \leq c(y_1, y_2, \dots, y_t), \tag{11}$$

where $c(y_1, y_2, \dots, y_t) = \displaystyle\sum_{i=1}^{t} \pi_i\ y_i + \frac{\pi_{t+1}}{w_{t+1}} \left(W_{max} - \sum_{i=1}^{t} w_i\ y_i \right)$.

Furthermore given $M \in \mathbb{R}$ such that $c(y_1, y_2, \dots, y_t) \leq M$, it holds:

$$g(z) \leq M \quad \textit{for all} \quad z \in \mathcal{A} \quad \textit{such that} \quad \left\{\begin{array}{l} z_i = y_i, \quad i = 1, \dots, t-1, \\ z_t = y_t - \alpha, \end{array}\right. \tag{12}$$

with $\alpha > 0$.

In the previous paragraphs the constraint $w^T\ y \leq W_{max}$ has priority over the cardinality constraint: $\displaystyle\sum_{i=1}^{n} y_i \leq N-1$. However, if N and w_i are small, it is advisable to exchange the role of those constraints to get tighter bounds on the objective function values. In our implementation we allow to exchange the role of the constraints. To give priority to the cardinality constraint the variables are arranged such that the associated efficiencies are in decreasing order: $\pi_1 \geq \pi_2 \geq \dots \geq \pi_n$, and $c(y_1, y_2, \dots, y_t)$ is redefined by

$$c(y_1, y_2, \dots, y_t) = \sum_{i=1}^{t} \pi_i\ y_i + \pi_{t+1} \left(N - 1 - \sum_{i=1}^{t} y_i \right).$$

Another strategy to save computations is to consider a good upper bound U^{max} on the optimal solution value. Hence we can stop the search process when we find a feasible point y verifying $g(y) = U^{max}$. Before describing our upper bound some notation is needed. We consider two relaxations of problem (9): in (P_1) the cardinality constraint is discarded and in (P_2) the constraint $W_{min} \leq w^T y \leq W_{max}$ is eliminated. The following result can be obtained using a similar argument to that of Theorem 2.2.1 given in [7] for 0-1 Knapsack problems.

Proposition 3 *If the variables of problem (P_1) are sorted by decreasing efficiencies (10), then a global solution of (P_1) is given by:*

$$\tilde{y}_i = \begin{cases} u_i, & 0 < i \leq s-1, \\ \dfrac{W^R_{max}}{w_s}, & i = s, \\ 0, & s < i \leq n, \end{cases}$$

where $s = \min\left\{t \in \{1,\ldots,n\} : \sum_{j=1}^{t} w_j\ u_j > W_{max}\right\}$ *and* $W^R_{max} = W_{max} - \sum_{j=1}^{s-1} w_j\ u_j$.

As a consequence of Proposition 3 the following result can be deduced (see [3] for the proof):

Corollary 1 *An upper bound on $g(\tilde{y})$ is given by $U = \max(U_1, U_2)$ with*

$$U_1 = \sum_{\substack{i=1 \\ i \neq s}}^{\bar{s}-1} \pi_i\ u_i + \pi_s\ \beta_s + \bar{W}^R_{max}\ \frac{\pi_{\bar{s}}}{w_{\bar{s}}} \quad \text{and} \quad U_2 = \sum_{i=1}^{\tilde{s}-1} \pi_i\ u_i + \pi_s\ (\beta_s + 1) + \tilde{W}^R_{max}\ \frac{\pi_{\tilde{s}}}{w_{\tilde{s}}},$$

where s and W^R_{max} are those of Proposition 3, $\beta_s = \left\lfloor \frac{W^R_{max}}{w_s} \right\rfloor$,

$$\bar{W}^R_{max} = W_{max} - w_s\ \beta_s - \sum_{\substack{i=1 \\ i \neq s}}^{\bar{s}-1} w_i\ u_i, \quad \tilde{W}^R_{max} = W_{max} - w_s\ (\beta_s + 1) - \sum_{i=1}^{\tilde{s}-1} w_i\ u_i,$$

$$\bar{s} = \min\left\{t \in \{s+1,\ldots,n\} : \sum_{\substack{i=1 \\ i \neq s}}^{t} w_i\ u_i > W_{max} - w_s\ \beta_s\right\}$$

$$\text{and} \quad \tilde{s} = \min\left\{t \in \{1,\ldots,s-1\} : \sum_{i=1}^{t} w_i\ u_i > W_{max} - w_s\ (\beta_s + 1)\right\}.$$

Martello and Toth [8] proposed the upper bound given by $U^{MT} = \max\{U_1^{MT}, U_2^{MT}\}$ with

$$U_1^{MT} = \sum_{i=1}^{s-1} \pi_i\ u_i + \pi_s\ \beta_s + \bar{W}^{RMT}_{max}\ \frac{\pi_{s+1}}{w_{s+1}}, \quad U_2^{MT} = \sum_{i=1}^{s-2} \pi_i\ u_i + \pi_s\ (\beta_s + 1) + \tilde{W}^{RMT}_{max}\ \frac{\pi_{s-1}}{w_{s-1}},$$

where s and β_s defined in Proposition 3 and Corollary 1, resp.,

$$\bar{W}^{RMT}_{max} = W_{max} - w_s\ \beta_s - \sum_{i=1}^{s-1} w_i\ u_i \quad \text{and} \quad \tilde{W}^{RMT}_{max} = W_{max} - w_s\ (\beta_s + 1) - \sum_{i=1}^{s-2} w_i\ u_i.$$

It can be proved that our bound is tighter than theirs ([3]):

Corollary 2 *The upper bounds satisfy $U \leq U^{MT}$.*

On the other hand, it is clear that U^{MT} need less computations; moreover, both coincide ($U = U^{MT}$), when $\bar{s} = s + 1$ and $\tilde{s} = s - 1$. Using once more Proposition 3, we obtain the optimal solution value of knapsack problem (P_2).

Proposition 4 *If the variables of problem* (P_2) *are sorted by decreasing efficiencies:* $\pi_1 \geq \pi_2 \geq \cdots \geq \pi_n$*, then a global solution of* (P_2) *is defined as follows:*

$$\hat{y}_i = \begin{cases} u_i, & 0 < i \leq \hat{s} - 1, \\ C_N, & i = \hat{s}, \\ 0, & \hat{s} < i \leq n, \end{cases}$$

where $\hat{s} = \min\left\{t \in \{1, \ldots, n\} : \sum_{j=1}^{t} u_j > N - 1\right\}$ *and* $C_N = N - 1 - \sum_{j=1}^{\hat{s}-1} u_j$.

Combining previous results, we arrive to the following bound on the optimal solution value of problem (9), (see [3] for the proof):

Theorem 2 *An upper bound on* $g(\bar{y})$ *is given by* $U^{max} = \min(U, g(\hat{y}))$.

3.3. Algorithm for Solving the Knapsack Problem (9)

Step 0 Compute the upper bound U^{max} for the optimal solution value. Choose the main constraint to arrange the variables such that the associated efficiencies be in decreasing order. Set to zero the index of the branching variable, $k = 0$, the current best solution, $y^* = 0$, and its objective function value, $g^* = 0$.

Step 1 Find the most promising extension of the current branch:

- **1a.** Update $k = k + 1$ and set $y_k = \min\left(\left\lfloor \frac{W_{max} - \sum_{j=1}^{i-1} w_j\ y_j}{w_i} \right\rfloor, u_i, N - 1 - \sum_{j=1}^{i-1} y_j\right)$.
- **1b.** If $k < n$ and $c(y_1, \ldots, y_k) > g^*$ go to **1a**; else, for $i = k + 1$ to n set $y_i = 0$.

Step 2 If y is feasible and $g(y) > g^*$, then replace y^* by y and g^* by $g(y)$. If $g^* < U^{max}$, then go to Step 3; else, stop.

Step 3 Backtrack to the next branch:

- **3a.** if $k = 1$, then stop; else update $k = k - 1$;
- **3b.** If $y_k > 0$, then update $y_k = y_k - 1$ and go to **1b**; else go to **3a**.

3.4. Column Generation Phase I. Initial pattern set

We consider two procedures for choosing the patterns of the first restricted LP problem.

GG We start with n disjoint patterns, each one containing the largest possible integer number of a unique ordered size, following the approach of Gilmore and Gomory [5]. Hence, for $j = 1, ..., n$ we define

$$a_{ij} = \begin{cases} \min\left\{d_i, N-1, \left\lfloor \frac{W_{max}}{w_i} \right\rfloor\right\} & if \quad i = j \\ 0 & otherwise. \end{cases}$$

FFD The largest ordered width is assigned to the first pattern with enough remaining capacity by applying a first fit decreasing algorithm. Assuming the widths are arranged in decreasing order, the iteration begins with certain residual demands $(d_1^r, d_2^r, \ldots, d_n^r)$ and we define the cutting pattern by

$$a_{ij} = \begin{cases} \min\left\{d_1^r, N-1, \left\lfloor \frac{W_{max}}{w_i} \right\rfloor\right\} & for \quad i = 1, \\ \min\left\{d_i^r, N-1, \left\lfloor \frac{W_{max} - \sum_{k=1}^{i-1} w_k\ a_{kj}}{w_i} \right\rfloor\right\} & for \quad i = 2, \ldots, n. \end{cases}$$

Then we replace d_i^r by $d_i^r - t_j\ a_{ij}$ where t_j is the number of times pattern j is used.

Note that these patterns may not satisfy the lower bound constraint on the size of the pattern. Our algorithm eliminates this type of patterns if they do not contribute to an intermediate LP solution.

3.5. Column generation. Phase III

Here we have an optimal LP solution, $\bar{x}^{LP}$, and the corresponding set of patterns. The integer problem corresponding to this set of patterns can be infeasible. One way of avoiding this difficulty is to add new patterns. The FFD algorithm is used now to complete the set of patterns before solving the restricted integer problem. We add sufficient patterns to guarantee that a feasible point exists, starting with the residual demands

$$d_i^R = \max \left\{ 0, d_i - \sum_{j=1}^{m_R} \bar{a}_{ij}\ \bar{x}_j \right\},\ i = 1, \ldots, n,$$

where m_R is the number of patterns after solving the LP relaxation. At the end of this process we also get an integer feasible point, x^0, and therefore an upper bound for the integer problem, $f(x^0)$.

3.6. Main CSP Algorithm

Step 0 The choice of the optimization model and the pattern generation system are given, as well as the over-production tolerances.

Step 1 Preprocessing of the data: eliminate duplicate item sizes by grouping the corresponding demands.

Step 2 Pattern generation I: compute the initial pattern set and the data of the initial restricted problem.

Step 3 LP Phase.

- **Step 3a.** Call the LP algorithm for solving the LP restricted problem. It provides a solution x^{LP} and its Lagrange multipliers.
- **Step 3b** Pattern generation II: Call the Knapsack algorithm. It provides a new pattern a.

  Eliminate the initial patterns that violate the minimum width and are not used by x^{LP}.

  If (7) holds, increase the pattern set with the new pattern a and go to Step 3a.

Step 4 Rounding down the non-integer components of x^{LP}. If $\lfloor x^{LP} \rfloor$ is feasible for the integer problem associated to the last LP problem, go to Step 6.

Step 5 Pattern generation III. Add enough patterns to guarantee that a feasible integer point exists.

Step 6 Solve the final restricted integer problem.

4. Numerical Results

We have implemented the algorithms described above in MATLAB linked with LINDO API on a personal computer with 2.66 GHz and 2 GB of memory, running under Windows XP. We have called LINDO API for solving the restricted LP problems with a simplex method and the final restricted integer problem with a branch and cut method.

The main purpose of the computational tests was to validate the algorithm with industrial instances given by the company, compare the two models proposed above and see the relative impact of GG and FFD column generation procedures. Here we present a selection of the numerical results obtained with data sets from paper industry and 60 randomly generated data sets. In both cases we have worked with over-production given by $D_i = \max\{\lfloor 1.05\ d_i \rfloor, d_i + 1\}$ units for $i = 1, \ldots, n$.

Table 1 summarizes the characteristics of the real cutting stock data sets and some numerical results for these examples. The first column gives the data set identification number. For each instance we show the number of finished roll types, n, the range of the demanded widths, $Range_w$, the average width, $Aver_w = \left(\sum_{i=1}^{n} d_i w_i \right) \Big/ \left(\sum_{i=1}^{n} d_i \right)$, $Range_d$ and $Aver_d$ give the range of the demanded units for any width and the average demand per width. The last four columns present numerical results obtained by using FFD procedure in Phase I and Phase III of column generation $(FFD + FFD)$: the percentage of trim loss, TL, and the total number of used master rolls, NMR, for Model I and Model II. The minimum and maximum widths, W_{min} and W_{max}, have been set to 375 and 430 units, respectively. The number of available knives is $N = 11$ for this company. These problems seem to be small because the number of ordered widths range from 3 to 24; however, they represent the usual production orders in the company.

Table 1: Numerical results with $FFD + FFD$ for paper industry problems.

	Data Set Characteristics					Model I		Model II	
DS	n	$Range_w$	$Aver_w$	$Range_d$	$Aver_d$	TL (%) .	NMR	TL (%).	NMR
1	13	60.0 - 133	104.1	3 - 44	16	$*$ 0	52	1.5	• 52
2	9	46.3 - 141.3	90.7	3 - 78	25.4	3.2	50	3.4	50
3	8	24.0 - 132.5	100.7	4 - 96	35	0.9	67	2.1	67
4	8	60.0 - 129	100.9	4 - 85	20.6	0.6	40	0.7	39
5	9	25.0 - 202	112.1	1 - 51	14.4	2.6	36	3.5	36
6	12	25.0 - 202	113.6	1 - 62	17.6	3.2	60	3.9	58
7	24	24.0 - 202	105.9	1 - 96	23.1	0.1	140	0.3	• 138
8	24	50.5 - 133	109.4	3 - 87	14.5	0.8	91	1.9	90
9	5	90.5 - 142.5	119.1	8 - 27	15.6	$*$ 6.1	23	6.1	23
10	5	90.0 - 141.3	104.9	7 - 34	16.4	2.6	21	4.8	21
11	3	63.5 - 126.5	116	8 - 62	27	8.9	24	8.9	24
12	5	87.3 - 141.3	122.2	3 - 51	21.8	5.5	33	6.1	33
13	3	37.4 - 63.3	42.2	5 - 29	13.3	1.9	4	1.9	4
14	4	25.0 - 202	60	1 - 17	5.3	1.5	4	2.2	3
15	3	63.5 - 134.5	106.5	4 - 11	6.3	5.9	5	5.9	5
16	4	59.5 - 141.5	122.2	4 - 16	12.3	$*$ 6.2	15	6.2	15
17	4	57.3 - 122.5	102.5	6 - 28	16	2	16	2.9	16
18	4	66.5 - 132.1	88.9	2 - 16	6.5	10.4	6	10.4	6

With Model II the optimal solution value was reached for 16 of the 18 examples. A black point in the last column indicates that the computed integer solution value and the optimal integer function value differ one unit. On the other hand, for Model I, an asterisk indicates that the gap between the LP solution and the IP solution is zero.
Figure 1 presents the solution qualities obtained with Model I and Model II for the first twelve instances described in Table 1.

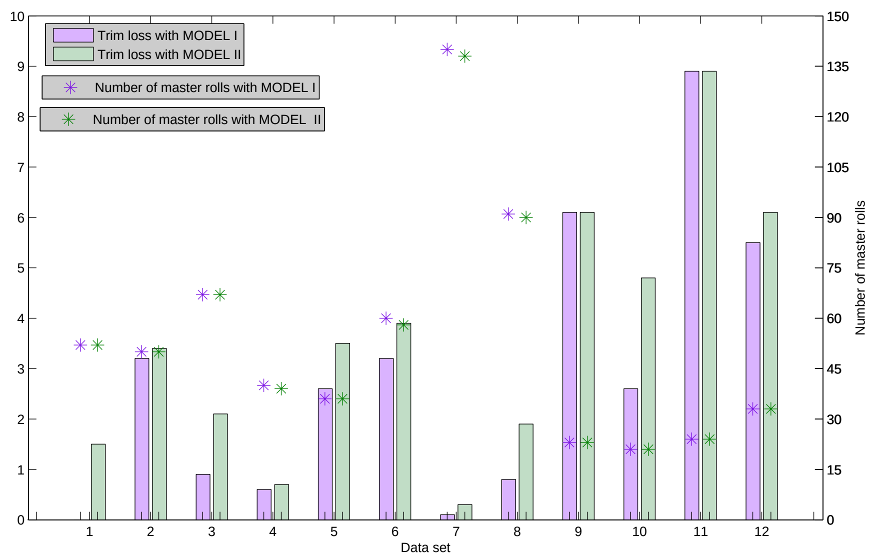

Figure 1: Comparison of models.

Table 2: Numerical results with $GG + FFD$ for paper industry problems.

	Model I		Model II	
DS	TL (%)	NMR	TL (%)	NMR
1	0	52	1.2	$*$ 51
4	0.6	40	3.2	40
5	3	36	4.3	36
6	3.2	60	3.9	58
7	0.1	140	0.7	138
14	2.2	3	2.2	3
17	2	16	4.6	16

In Table 2 we give the numerical results obtained by using GG procedure in Phase I and FFD mechanism in Phase III of column generation ($GG + FFD$). We only present the results that differ in the quality of the solutions from the previous ones. An asterisk marks the only one case where $GG+FFD$ was superior to $FFD + FFD$. Let us remark that $GG + FFD$ consumes more computing time than $FFD + FFD$, specially until LP Phase is finished.

Finally, to evaluate the influence of the number of items we have worked with examples generated randomly using CUTGEN1 [4]. The size of the problem, n, has been fixed to $n = 10, 20, 30$ and 40. The bounds on the master roll width, $W_{max} = 430$, $W_{min} = 375$ and the number of knives, $N = 11$, have the same values as above. The average demand per width was 25, and the lower and upper bounds for the relative size of order lengths in relation to W_{max} were 0.1 and 0.3, respectively. We have selected these values to produce random examples having similar characteristics than the industrial ones cited previously. For each class 15 problem instances have been generated. Tables 3 and 4 report numerical results with $FFD + FFD$ and $GG + FFD$, respectively, showing average results over the data sets. For each group of examples the column “LP Time” (“IP Time”) gives the average CPU time (in seconds) consumed during the LP phase (resp., final IP problem). Looking at the average trim loss (TL column), it is clear that the quality of the results is very good for both models (specially for Model I), even better than in Table 1, for the examples given by the company. It can be also seen how the “LP” and “IP” times increase with the size of the problem, n, as expected, and that $GG + FFD$ consumes more computing time than $FFD + FFD$.

Table 3: Numerical results with $FFD + FFD$ for random problems.

n	Model I				Model II			
	LP Time (sec)	IP Time (sec)	TL (%)	NMR	LP Time (sec)	IP Time (sec)	TL (%)	NMR
10	6.09	0.53	0.39	53.07	3.39	0.25	1.62	52.87
20	10.93	0.76	0.01	105.80	4.44	0.65	0.81	104.80
30	12.92	1.16	0.01	153.60	5.80	$*$ 8.09	0.59	151.60
40	14.80	2.25	0	201.07	9.57	11.33	0.47	198.13

Table 4: Numerical results with $GG + FFD$ for random problems.

n	Model I				Model II			
	LP Time (sec)	IP Time (sec)	TL (%)	NMR	LP Time (sec)	IP Time (sec)	TL (%)	NMR
10	7.50	0.52	0.41	53.27	5.37	0.23	1.51	52.80
20	15.96	0.69	0.01	106	10.13	0.79	0.91	104.93
30	19.51	1.07	0.02	153.60	14.07	1.78	0.66	151.67
40	24.19	1.47	0	201.20	19.74	$*$ 51.20	0.50	198.13

For Model II, an asterisk indicates that only one example of the corresponding class ($n = 30$ or $n = 40$) clearly deviates from the average “IP Time” (about eleven times that value).

6. Conclusions

We have considered two models for the one-dimensional cutting stock problem. From a practical standpoint, Model I allows to compute an optimal solution with less trim loss than Model II by a small increase in the number of master rolls in the worst case. The objective function value obtained by solving Model II is a good reference point for evaluating the quality of the solution obtained with Model I. Finally, we conclude that column generation procedure FFD is superior to GG approach, comparing computing times and working with random data sets. Moreover, the results attained for the paper industry examples also show a clear difference for the computing times, having a similar quality for the results.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation under grant No. $MTM2008-04206$.

References

[1] H. Ben Amor and J.M. Valério de Carvalho, Cutting Stock Problems, *Column Generation*, G. Desaulniers, J. Desrosiers and M.M. Solomon (Eds.), Springer, New York, 2005.

[2] V. Chvátal, *Linear programming*, W.H. Freeman, New York,1983.

[3] L. Fernández, *Técnicas matemáticas para el problema de cortado*, Trabajo Dirigido en Estadística y Computación, Licenciatura de Matemáticas, Universidad de Cantabria, 2010.

[4] T. Gau and G. Wäscher, CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem, *European Journal of Operational Research*, 84 (3), 572-579, 1995.

[5] P.C. Gilmore and R.E. Gomory, A linear programming approach to the cutting-stock problem. *Operations Research*, 9, 849–859, 1961.

[6] P.C. Gilmore and R.E. Gomory, A linear programming approach to the cutting-stock problem - Part II. *Operations Research*, 11 (6), 863–888, 1963.

[7] H. Kellerer and U. Pferschy and D. Pisinger, *Knapsack problems*, Springer-Verlag, Berlin, 2004.

[8] S. Martello and P. Toth, *Knapsack problems*, John Wiley & Sons Ltd., Chichester, 1990.

[9] J.F. Pierce, *Some large-scale production scheduling problems in the paper industry*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[10] G. Scheithauer and J. Terno, The modified integer round-up property for the one-dimensional cutting stock problem. *European Journal of Operational Research*, 84 (3), 562-571, 1995.

[11] G. Wäscher and H. Haußner and H. Schumann, An improved typology of cutting and packing problems, *European Journal of Operational Research*, 183 (3), 1109–1130, 2007.